A Semantic Communication Based Wireless Image Transmission for Internet of Things Devices

Rangang Zhu, Shengxian Huang, Chenguang He, Shaojing Su, and Hao Chen

Abstract In recent years, wildfires occur frequently as global warming. People set a large number of sensors to monitor the wild land. However, the poor network in remote area can hardly afford the big data transmission while low delay. In this paper, we firstly propose a semantic communication framework, which consists of the GAN-based semantic extraction and LDPC code. To make it affordable for Internet of Things (IoT) devices, we then compression the pre-trained model by parameters pruning and clustering with the acceptable price of inference performance. Based on our analysis, the proposed semantic communication system can significantly reduce the volume of transmission data by extracting the semantic information of images and preform robustness in fading channel.

Keywords Semantic communication · Image compression · Internet of Things

1 Introduction

As global warming continues, extreme heat and dry weather occurs frequently, and the subsequent wildfire damage has got people's attention. People used to patrol the forest as guards, but now, a more common solution is that setting many sensors or edge devices to keep monitoring their surrounding in real time. Besides sending the warning signal in a faster and cheaper way, they can log more details and send them

R. Zhu
College of Electronic Engineering, National University of Defense Technology, Hunan, China

S. Huang · C. He (✉) · H. Chen
Communications Research Center, Harbin Institute of Technology, Harbin, China
e-mail: hechenguang@hit.edu.cn

S. Huang
e-mail: 21s105173@stu.hit.edu.cn

S. Su
College of Intelligence Science and Technology, National University of Defense Technology, Hunan, China

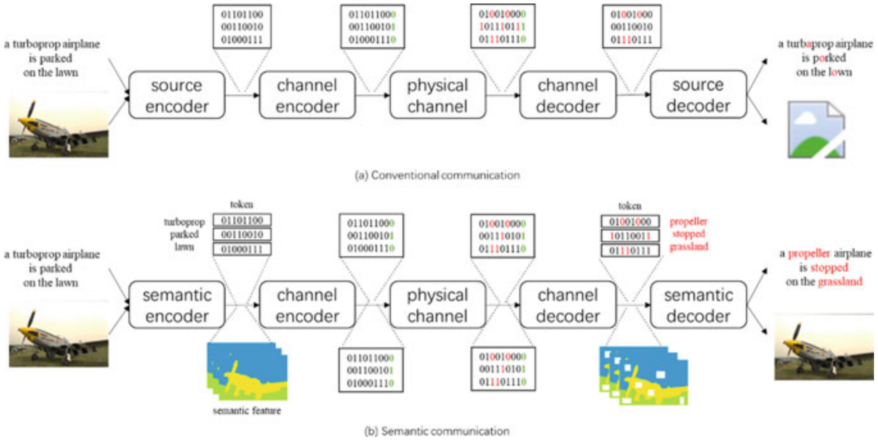
W. Wang et al. (eds.), *Communications, Signal Processing, and Systems*, Lecture Notes in Electrical Engineering 1032, https://doi.org/10.1007/978-981-99-7505-1_37

back to the center for further analysis. However, sensors are generally supposed to be energy-efficient and performance-limited. A sensor has a restricted service area, which means that it may need thousands of sensors to cover a hill. Although applying source compression, it is still a great burden for the transmission networks if lots of sensors send their data at the same time. Especially in some unusual cases like emergency rescue and damage assessment, live pictures or videos of interested area are always required.

To reduce the networks traffic and the probability of message collision, we need a more efficient image compression. However, conventional image compress like JPEG, BPG are regarded as high-efficiency engineering implementation to approximate the entropy of the image. Benefitting from the advancements of deep learning and end-to-end communication, semantic communication is promising to break the compression limit defined by Shannon information theory. Semantic communication system interprets received information at the semantic level rather than bit level, which is what we exactly do in convention communication system [1]. The difference between conventional communication and semantic communication application is illustrated in Fig. 1.

With the DL-based source coding and channel coding module, semantic features can be extracted and reconstructed quickly in high-performance computing environment. However, sensors can hardly afford the semantic interpreting processing of large semantic models. In this paper, we will focus on the lightweight semantic communication for edge devices.

There have been some initial works related to semantic communication and lite semantic communication for image transmission. [2, 3] proposed efficient joint source-channel coding methods for wireless image transmission based on the convolution neural network (CNN), respectively named DeepJSCC and DeepJSCC-f, where the latter firstly exploited the channel output feedback in training and surpass



(a) Conventional communication

(b) Semantic communication

Fig. 1 Conventional communication and semantic communication

the traditional structured coding-based designs. However, the compression rate seems to be constrained by the bottleneck of CNN, lacking of the comparison with advanced image compression code. A lite semantic communication model for limited computing capability IoT devices is discussed in [4]. It proposed a low complexity text transmission model based on transformer and developed a channel state information (CSI) aided training processing to promise IoT devices to get the correct data and train the distributed model locally, but it is difficult to design a lite image transmission model which prefers convolution layers rather than dense layers.

2 System Model

Referencing to the block-based design in conventional communication system, we take the semantic communication system apart into two modules. One is for semantic feature extraction and the other is for semantic feature transmission. Observed that wireless channels in physical world change stochastically, and the random data makes the model training difficult to converge. There has a paradox that the data driven model usually requires the correct input for a more accurate interpretation which means DL-based channel coding module needs a huge number of parameters to learn statistical characteristics of channel. It significantly increases the model complexity and becomes unaffordable for IoT devices, otherwise model cannot provide helpful semantic features for convergence. The structure of semantic transmission system is shown in Fig. 2.

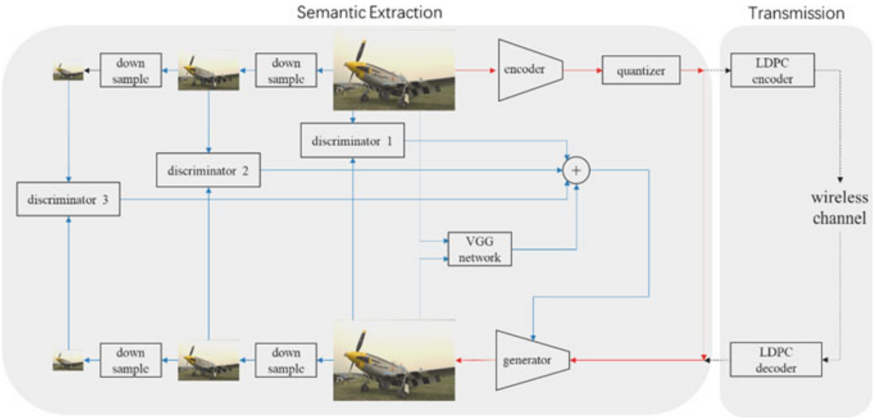


Fig. 2 System structure

2.1 Transmission Module

To simplify the semantic communication system, we adapted LDPC coding to keep the semantic information high fidelity while transmission. LDPC coding is a mature scheme and widely used in 5G. It provides a high efficiency and high reliability channel coding within low complexity, which can be implemented at low hardware cost.

2.2 Semantic Extraction Module

Once we solved the effect of fading channel with low consumption, more computing resource of IoT devices can be allocated into running a DL model. The semantic extraction module can concentrate on image extreme compression. The target of semantic extraction model can be modeled after the semantic rate-perception-distortion theory [5], which gives the limit of semantic information rate R within the conventional average distortion D and semantic perceptual distortion P, i.e.,

$$R(D, P) = \min_{p(\hat{s}|s)} I\big(s, \hat{s}\big) \quad \text{s.t.} \quad \mathbb{E}\big[\Delta\big(s, \hat{s}\big)\big] \le D, d[p_s, p_{\hat{s}}] \le P \tag{1}$$

where $I\big(s, \hat{s}\big)$ is the mutual semantic information between the transmitter s and the receiver $\hat{s}$. Since the divergence distance is considered as an effective index of semantic perceptual quality, we define the divergence distance between distributions as $d[\cdot, \cdot]$. GANs [6], which is firstly proposed by Goodfellow, has been demonstrating superior performance than CNNs on image application like generation, reconstruction and so on. The structure of alternately training the generator $G(\cdot)$ and the discriminator $D(\cdot)$ for a saddle point of min–max objective with the loss function, which is formulated as

$$L_{\text{GAN}}(G, D) = \max_{D} \mathbb{E}[f(D(s))] + \mathbb{E}[g(D(G(z)))] \tag{2}$$

is proven to be equivalent to measure the divergence distance between the probability distribution of the origin dataset and the generated dataset. The proposed training structure is based on the Least-Squares GAN [7], which corresponds to the Pearson χ^2 divergence with $f(x) = (x - 1)^2$ and $g(x) = x^2$.

We adapt a multi-scale discriminator, which consisted of three independent and identical PatchGANs [8], to measure the semantic features loss of the origin image and its downsampled images. The quantizer q maps the encoder output from float-point number to integer. Moreover, we adopt VGG loss to navigate the model to generate low average distortion images. The optimization objective can be formulated as

$$\min_{E,G}\max_{d_k \in D}\left[\sum_{k=1}^{N_D} L_{\mathrm{LSGAN}}(G, d_k) + \lambda \mathbb{E}[\mathrm{VGGLoss}(s, G(q(E(s))))]\right] \tag{3}$$

where coefficient λ controls the rate of semantic perceptual distortion and conventional average distortion.

3 Model Compression and Acceleration

As revealed in the lottery ticket hypothesis [9], it is difficult to train a pruned model from start. A better solution is to prune while training. However, it is impracticable for performance-limited and power-limited IoT devices to retrain the pruned model within locally collected data. We tend to deploy the model one time and maintain the IoT devices if and only if necessary in the remote mountainous region with poor infrastructure. Although retraining with the backhaul data and redistributing the updated model is viable, the low bandwidth and high delay network might exhaust itself by such huge data transmission. Hence, a plug and play IoT device with a pre-compression is better suited for wildfire monitoring scenario. Considering that the IoT devices usually work in a preset position even the cameras face toward a fixed direction and we can deal with the trade-off between accuracy and convenience in a simpler way. In this section, we will give out some general model compression and acceleration method. These all operations directly act on a pretraining large model without fine-tuning.

3.1 Parameter Quantification

The parameters of model are set to floating-point numbers by default, which needs 32 bits to save a single parameter (FLOAT32). The operation and storage of a large number of FLOAT32 are tolerable for compute unified device architecture (CUDA) supported computer. However, IoT devices are equipped with limited performance CPUs and without GPUs. They are supposed to handle integer data and provide low floating-point operations per second (FLOPS). Hence, we can quantize the parameter from FLOAT32 to INT8, and a general uniform quantization is as

$$\begin{aligned}
&Q=\frac{R}{S}+Z\\
&R=(Q-Z)\times S\\
&S=\frac{R_{\max}-R_{\min}}{Q_{\max}-Q_{\min}}\\
&Z=Q_{\max}-\frac{R_{\max}}{S}
\end{aligned} \tag{4}$$

where Q, R represent the quantized value and real value, respectively. S is the scale of quantization and Z is the zero point. Z is usually equal to 0 exactly because it plays an important role in the whole model, which deserves a special treatment. Furthermore, to quantize the whole model, additional operation should be applicated. We firstly took several typical valves of dataset to translate the type of the input and output. Then we reinterpreted the model with the filtered dataset to help the activation layer calibrate.

3.2 Parameter Pruning and Clustering

To find the potential winning ticket, the over-parameterization models are widely accepted. The redundant parameters can help handle corner cases and improve robustness while training, however, they become a burden for inference instead in a simple scenarios like wildfire monitoring. A portion of parameters in the over-parameterization model have minor effect on inference accuracy in fact, which can be replaced by their statistical characteristics without a heavy cost. Parameter pruning can be controlled by a hyper-parameter named sparsity, which is ranging from 0 to 1. All the parameters are sorted and the parameters below the threshold are set to 0 brutally. Parameter clustering is similar but gentler. It build a set of statistical characteristics of parameters and represent the parameters with their index in the set. The details are shown in Fig. 3.

Both pruning and clustering do not modify the network structure directly and they just give out the sparse representation of the model for further compression. Because the calculation of most DL model is based on the computation graphs, which are divided into dynamic graph and static graph. The difference between both is that the latter is invariant once the network have initialized. We prefer the dynamic graph for clearer debugging procedure while the static graph for higher execution efficiency.

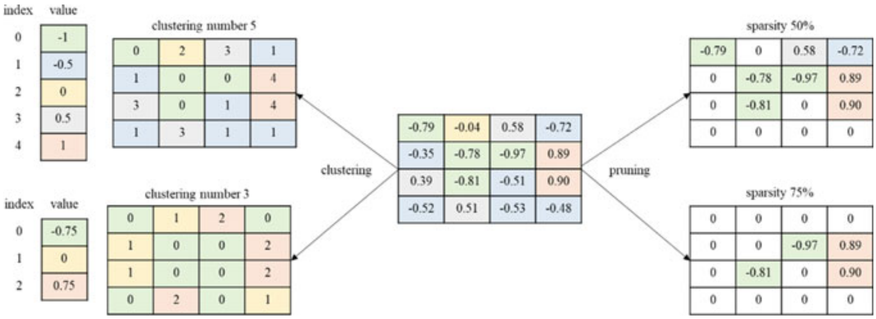


Fig. 3 Pruning and clustering

4 Numerical Results

In this section, we firstly compare the proposed semantic communication system with different conventional image compression. Then we compare the performance penalty of compressed model.

As shown in Fig. 4, the proposed semantic extraction module generated clearer images within less bit cost. Benefitting from the generator, which interprets and reconstructs with the received semantic information, the volume of data reduce significantly. Unlike the conventional image compression, they treat the every pixel of images equally and reconstruct blur results. The proposed model keeps a good visual perception separately in both the subject and the background of images but the border, which can be demonstrated from the difference of the flame shape. Figure 5 shows the compression performance of ours module and conventional compression. We can see that.

Figure 6 shows the transmission performance between the proposed system and DeepJSCC over the AWGN channel and Rayleigh channel. DeepJSCC performed better in low SNR environment with the additional training against channel fading. However, the fading effect is simulated by the complex Gaussion distribution and it is difficult to apply the theoretical results into practice. The semantic transmission system interpreted negative results from the incorrect semantic information but its

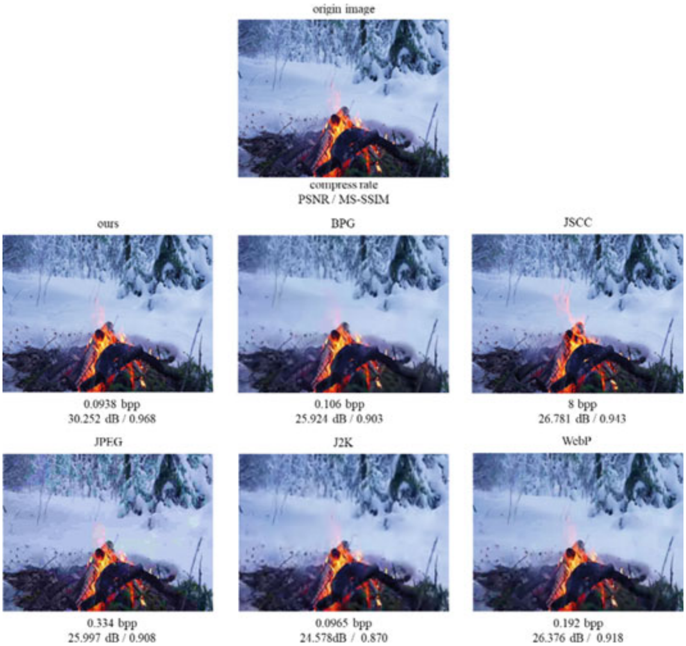


Fig. 4 Example of visual comparison

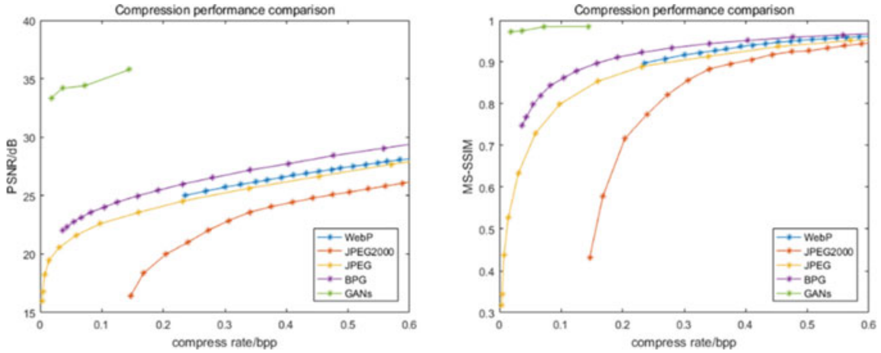


Fig. 5 Compression performance comparison

performance increases rapidly with as the SNR increases owing to the robustness of LDPC code.

Table 1 shows the result of the proposed model before and after the compression. Noticed that the number of parameters is constant because of the computation graph mentioned above. And the loss of the generated results are acceptable, where the

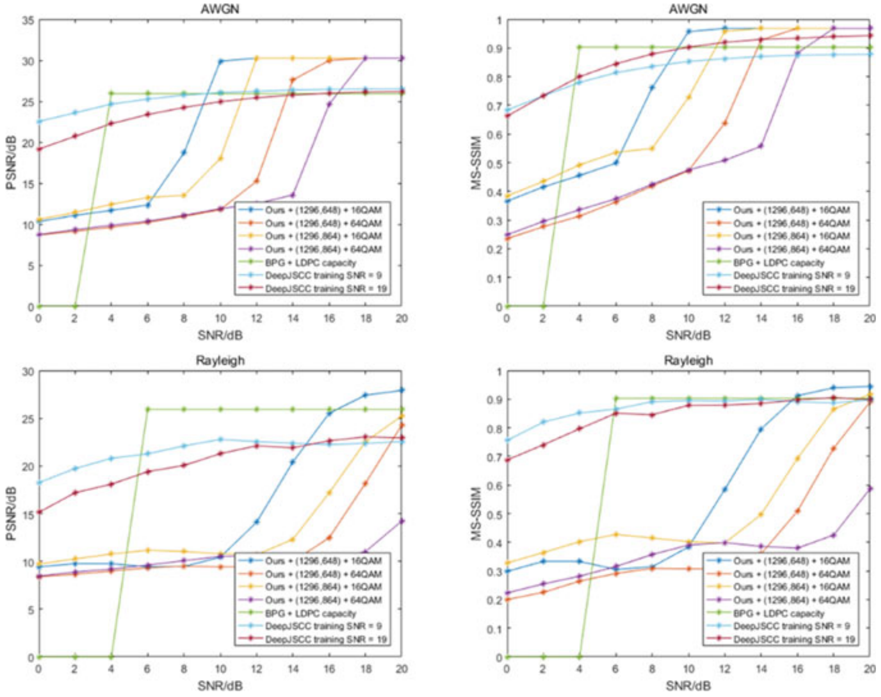


Fig. 6 Transmission comparison, where LDPC denotes (block length, information length)

Table 1 Model compress and acceleration comparison

Quantization	measurement	Semantic extraction module				Pruning	Measurement	Semantic extraction module			
		Encoder		Generator				Encoder		Generator	
Weight only (int8)		Before	After	Before	After	Sparsity 50%		Before	After	Before	After
	Number of parameter	5,591,544	5,591,544	154,942,563	154,942,563		Number of parameter	5,591,544	5,591,544	154,942,563	154,942,563
	Model size	21.8 MB	5.35 MB	592 MB	141.37 MB		Model size	21.8 MB	19.51 MB	592 MB	544.87 MB
		Before		After				Before		After	
	PSNR	30.3 dB		29.1 dB			PSNR	30.3 dB		25.7 dB	
	MS-SSIM	0.97		0.94			MS-SSIM	0.97		0.92	
Parameter all (int8)		Before	After	Before	After	Sparsity 75%		Before	After	Before	After
	Number of parameter	5,591,544	5,591,544	154,942,563	154,942,563		Number of parameter	5,591,544	5,591,544	154,942,563	154,942,563
	Model size	21.8 MB	5.34 MB	592 MB	131.24 MB		Model size	21.8 MB	19.48 MB	592 MB	520.43 MB
		Before		After				Before		After	
	PSNR	30.3 dB		19.8 dB			PSNR	30.3 dB		23.8 dB	
	MS-SSIM	0.97		0.65			MS-SSIM	0.97		0.9	

Clustering	Measurement	Semantic extraction module			
		Encoder		Generator	
Num = 8		Before	After	Before	After
	Number of parameter	5,591,544	5,591,544	154,942,563	154,942,563
	Model size	21.8 MB	2.89 MB	592 MB	80.01 MB
		Before		After	
	PSNR	30.3 dB		26.6 dB	
	MS-SSIM	0.97		0.93	

(continued)

Table 1 (continued)

Clustering	Measurement	Semantic extraction module			
		Encoder		Generator	
Num = 3		Before	After	Before	After
	Number of parameter	5,591,544	5,591,544	154,942,563	154,942,563
	Model size	21.8 MB	1.65 MB	592 MB	30.32 MB
		Before		After	
	PSNR	30.3 dB		23.4 dB	
	MS-SSIM	0.97		0.9	

Table 2 Runtime comparison

Model	Encode time/s	Decode time/s	Compress rate/bpp
JSCC	0.0235	1.027	8
Ours	1.2766	4.524	0.094
JPEG	0.0498	0.0173	1.416
JPEG2000	0.2717	0.1916	1.036
BPG	1.9982	0.3122	0.755

average of PSNR and MS-SSIM are above 25 dB and 0.9, respectively. The result of model size may be counterintuitive because the parameters consist of weights and bias and we found that the bias make a significant impact while interpreting, which can be proofed by the result of all parameters quantization. Thus we only operated on the weights of parameters. Table 2 shows the speed of our system and the others, where we took PSNR 30 dB as the benchmark with the Raspberry Pi 4B. We see that the proposed compressed model spends more time on semantic coding due to the limit parallel computing power. However, considering the poor network in the wilderness, we should pay more attention to reducing the transmission delay with the higher compression rate. In such scenario, a feasible structure is that the IoT devices encode and send the semantic information while the center completes the semantic reconstruction. Besides, we performed the simulation again in GPU by the computer with NVIDIA GeForce RTX 2080 Ti, and it took 0.7 s to complete the semantic reconstruction of one frame.

5 Conclusion

In this paper, we proposed an available semantic communication for IoT devices, which can work in a limit computing capabilities environment. Unlike the DeepJSCC, we considered the design of source semantic coding and channel coding separately. The former, which is based on the LSGAN, helped image extreme compression by extracting the semantic information while the latter provided an efficient and inexpensive method to keep the semantic information low distortion over the fading channel. To avoid the model retraining and the data backhaul, we compress the parameters of the pre-trained model directly by quantization, pruning and clustering. The simulation result demonstrated that the proposed semantic communication provided a higher compress rate and better reconstruction than other systems within comparable runtime.

References

1. Luo X, Chen H-H, Guo Q (2022) Semantic communications: overview, open issues, and future research directions. IEEE Wirel Commun 29(1):210–219. https://doi.org/10.1109/MWC.101.2100269
2. Bourtsoulatze E, Burth Kurka D, Gündüz D (2019) Deep joint source-channel coding for wireless image transmission. IEEE Trans Cognitive Commun Netw 5(3):567–579. https://doi.org/10.1109/TCCN.2019.2919300
3. Kurka DB, Gündüz D (2020) DeepJSCC-f: deep joint source-channel coding of images with feedback. IEEE J Selected Areas Inf Theor 1(1):178–193. https://doi.org/10.1109/JSAIT.2020.2987203
4. Xie H, Qin Z (2021) A lite distributed semantic communication system for Internet of Things. IEEE J Sel Areas Commun 39(1):142–153. https://doi.org/10.1109/JSAC.2020.3036968
5. Blau Y, Michaeli T (2018) The perception-distortion tradeoff. In: 2018 IEEE/CVF conference on computer vision and pattern recognition, Salt Lake City, UT, USA, pp 6228–6237. https://doi.org/10.1109/CVPR.2018.00652
6. Goodfellow I, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, Courville A, Bengio Y (2014) Generative adversarial nets. In: Advances in neural information processing systems, pp 2672–2680
7. Mao X, Li Q, Xie H, Lau RYK, Wang Z, Smolley SP (2017) Least squares generative adversarial networks. In: IEEE international conference on computer vision (ICCV). IEEE, pp 2813–2821
8. Isola P, Zhu J-Y, Zhou T, Efros AA (2017) Image-to-image translation with conditional adversarial networks. In: 2017 IEEE conference on computer vision and pattern recognition (CVPR), Honolulu, HI, USA, pp 5967–5976. https://doi.org/10.1109/CVPR.2017.632
9. Frankle J, Carbin M (2019) The lottery ticket hypothesis: finding sparse, trainable neural networks. In: 7th international conference on learning representations, ICLR 2019, New Orleans, LA, USA, 6–9 May 2019, pp 1–42